Project Proposal: SLLIM - System Log Local Intelligent Model

Authors: Carlos Cruzportillo, Nassos Galiopoulos, Jason Gillette

Affiliation: University of Texas at San Antonio Department of Information Technology and Cyber Security

Date: October 7th, 2024

potential security threats, diagnosing system issues, and ensuring compliance in real-time has become increasingly difficult due to the overwhelming amount of log data. This complexity demands advanced tools that can intelligently process, analyze, and query logs efficiently and cost-effectively. Moreover, the tools must be lightweight and adaptable, ensuring they can be deployed across a range of environments without introducing significant resource overhead or latency.

Problem Statement: The increasing volume of system logs generated by interconnected devices and enterprise systems create a challenge for IT professionals in efficiently detecting threats and diagnosing issues, necessitating the development of lightweight, intelligent tools for real-time log analysis and query.

The proliferation of interconnected devices and enterprise systems has exacerbated the challenge of managing and securing IT infrastructure. As logs from these systems increase in both volume and complexity, IT professionals are faced with a significant data management problem,

Research Questions

1. How well can a lightweight large language models (LLMs) detect system issues and security threats from system logs?
2. How effectively can lightweight large language models perform question answering tasks compared to larger, more resource-intensive models?

Specific Objectives

1. Fine-tune at least (2) lightweight large language models for comparative analysis.
2. Evaluate question answering performance of lightweight large language models against more popular and resource intensive large language models in the Cyber Security domain.

deploying lightweight LLMs on edge devices, particularly for real-time system log analysis. Mobile Edge Intelligence (MEI) uses edge networks with moderate computational resources to deploy LLMs closer to users, balancing privacy, latency, and computational load. This approach is crucial for developing a modular LLM architecture for effective on-device analysis.

3. **Paper 3**: The paper "Mobile Evaluation of Language Transformers" evaluates the feasibility of deploying LLMs on mobile and edge devices using their own benchmarking infrastructure called MELT. This study is particularly relevant for understanding the constraints and performance of lightweight LLMs on resource-constrained environments, which aligns directly with our objective of deploying modular, small-scale LLMs for offline forensic analysis.
4. **Paper 4**: The paper "LogQA: Question Answering in Unstructured Logs" introduces LogQA, a system designed to answer questions based on large-scale unstructured logs in natural language. This system addresses

fine-tuned by processing these prompts and comparing its output to the expected answers. Over iterative examples, the model learns to generate accurate and efficient responses to system log queries, helping IT professionals detect issues and threats more quickly.

```
graph TD;
    subgraph Training Data Flow
        A[Question Set] --> B[Question]
        C[Log Data] --> D[Context]
        B --> F[Prompt]
        D --> F
        E[Instruction] --> F
        F --> G[Model]
        H[Answer Set] --> G
    end
```

- **Evaluation Criteria**: For a comparative analysis of lightweight models versus a large language model, we will use a semantic similarity metric via a tool called BERTScore. BERTScore is a semantic evaluation mechanism used to assess the quality of generated text by comparing it to ground truth answer at the token level using contextual embeddings

scratch. Our ideal data includes system logs paired with relevant question-answer sets. These logs serve as the contextual foundation for training and fine-tuning our lightweight large language models to perform question answering tasks specific to cybersecurity and system issue detection. The ideal dataset would contain detailed log entries capturing various system events, along with question-answer pairs that reflect typical inquiries made by IT professionals, such as diagnosing system failures or identifying potential security threats. Each question will be fed to the model in a prompt with the coresponding log as context and an instruction for the model. By leveraging such a dataset, we can ensure that our models are not only able to understand and interpret system logs but also respond to specific questions accurately and efficiently. This dual-focus dataset is essential for both the training phase, where the model learns log-specific question answering, and the evaluation phase, where we measure model performance in addressing real-world system log queries. So far we have identified multiple datasets that contain a wide variety of system logs, but we have not yet

Expected Outcomes

In our comparative analysis of lightweight LLMs to larger models for the question answering task, we hypothesize that lightweight LLMs will deliver **faster inference** times and **consume fewer computational resources** while **maintaining competitive accuracy** for domain-specific tasks like system log analysis. While larger models possess vast parameter counts and a broader general knowledge base, our fine-tuned lightweight models will be optimized for question answering on system logs—requires; a more focused narrow-domain knowledge rather than vast, general knowledge.

Project Timeline

Milestone	Deadline
Proposal Submission	[20241007]
Dataset Collection	[20241014]
Model Training	[TBD]
Evaluation and Testing	[TBD]
Final Report Submission	[20241202]

the information in the logs, it could negatively impact the model's ability to train.

ii. Data Preparation: Managing the complexity of long and unstructured system logs requires careful preprocessing. Chunking large logs or using other data retrieval methods to extract the most relevant sections as prompt context will be critical. The challenge is ensuring that these extracted subsets are both representative and concise enough for efficient model input without losing essential information.

iii. Model Overfitting: There is a risk of the lightweight model overfitting to the domain-specific system logs during fine-tuning, which could cause it to lose its ability to generalize to new, unseen questions.

- **Mitigation Strategies**:
  i. Data Quality: To ensure alignment, implement strict human-in-the-loop validation that cross-check question-answer pairs against the log data.

  ii. Data Preparation: Develop an automated log chunking and retrieval

intelligence for large language models: A contemporary survey. IEEE Communications Surveys & Tutorials, 25(4), 2651-26903.

3. Gao, Y., Zhu, Y., & Zhu, H. (2023). Mobile evaluation of language transformers. Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 560-5704.
4. Liu, J., Chen, J., & Wang, X. (2023). LogQA: Question answering in unstructured logs. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 7892-79045.

TODO: Other papers and resources identified

- TinyAgent: Function Calling at the Edge
- On Device Language Models: A Comprehensive Review
- BERT-Log: Anomaly Detection for System Logs Based on Pre-trained Language Model
- CySecBERT: A Domain-Adapted Language Model for the Cybersecurity